Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

OPTIMIZADOR DE ACCESOS Y TRANSNMICIÓN DE DATOS

201901772 – Daniel Reginaldo Dubón Rodríguez

Resumen

El ensayo presenta la solución a un problema de distribución en una base de datos donde el costo total de acceso y transmisión de datos para el procesamiento es demasiado alto, se soluciona el problema obteniendo un nuevo esquema replicado de alojamiento adaptando un nuevo patrón de uso de la base de datos que minimiza los costos de acceso y transmisión, aplicando una metodología de agrupamiento, dado que los datos se pueden representar como una matriz de “n” filas y “m" columnas, el método consiste en obtener matrices de frecuencias de acceso y agrupar todas las tuplas que repitan el mismo patrón dando como resultado una matriz reducida de frecuencias de accesos, representando el nuevo esquema para el nuevo patrón de uso de la base de datos. Para la solución se optó por el uso de un lenguaje de programación para crear los algoritmos requeridos para realizar la metodología antes mencionada.

Palabras clave

Python, Tipo de Dato Abstracto, Lenguaje de Marcado Extensible, Programación Orientada a Objetos, Graphviz.

Abstract

The essay presents the solution to a distribution problem in a database where the total cost of access and data transmission for processing is too high, the problem is solved by obtaining a new replicated hosting scheme adapting a new pattern of use of the database that minimizes access and transmission costs, applying a grouping methodology, since the data can be represented as a matrix of “n” rows and “m” columns, the method consists of obtaining matrices of access frequencies and grouping all the tuples that repeat the same pattern resulting in a reduced matrix of access frequencies, representing the new scheme for the new pattern of use of the database. For the solution, the use of a programming language was chosen to create the algorithms required to carry out the aforementioned methodology.

Keywords

Python, Abstract Data Type, Extensible Markup Language, Objetct Oriented Programming, Graphviz.

Introducción

Se realizó un optimizador de accesos y transmisión de datos, con el cual se obtiene un nuevo esquema replicado de alojamiento adaptando un nuevo patrón de uso de la base de datos que minimiza los costos de acceso y transmisión, se realizó a través del lenguaje de programación Python para construir los diferentes algoritmos para obtener la solución así mismo se hizo uso del paradigma de programación orientada a objetos para abstraer el problema y lograr plasmarlo en el lenguaje de programación, también se usaron tipos de datos abstractos para crear una estructura en la cual se almacena los datos a procesar así mismo se hizo uso de la herramienta graphviz para crear el diseño de diagrama y generar una gráfica en el cual se pueden visualizar los datos obtenidos de un archivo con estructura XML.

Desarrollo del tema

El problema planteado consiste en alojar objetos de bases de datos en sitios distribuidos, de manera que el costo total de la transmisión de datos para el procesamiento de todas las aplicaciones sea minimizado.

Se ha optado el uso del lenguaje de programación Phyton debido a que es multiplataforma y por su facilidad de uso, además también es soporta varios paradigmas de programación.

Para la elaboración de la solución se provee un archivo con extensión y estructura xml, el cual trae información y modelo de los patrones de acceso que se desea optimizar y este archivo tiene la estructura de la siguiente forma:

```
<?xml version="1.0" ?>
<matrices>
    <matriz nombre="Ejemplo" n="#" m="#">
        <dato x="#" y="#">#</dato>
        <dato x="#" y="#">#</dato>
    </matriz>
</matrices>
```

Figura 1. Estructura del archivo XML dado.

Fuente: elaboración propia

donde:

matrices = es la etiqueta padre.

matriz = es la etiqueta que indica una nueva matriz creada para el respectivo análisis.

nombre = es el identificador de la matriz.

n = indica las filas que conformara la matriz.

m = indica las columnas que conformara la matriz

x = indica la fila de la matriz.

y = indica la columna de la matriz.

= indica un valor numérico

Obtenido ese formato se necesitaba de una estructura en la cual se guardarían la información del archivo dado, por lo cual se llevó al análisis del uso de un tipo de dato abstracto ya que permite el uso de memoria dinámica ya que el programa lo requiere debido a que no se sabe con certeza la cantidad de datos que puede contener.

Los tipos de datos abstractos implementados para la solución fueron, lista simplemente enlazada y lista enlazada circular.

En la lista simple enlazada se emulo una matriz que contendrá “n” filas y “m” columnas, para el almacenamiento de los datos, se optó por esta

estructura debido a su fácil implementación y utilización.

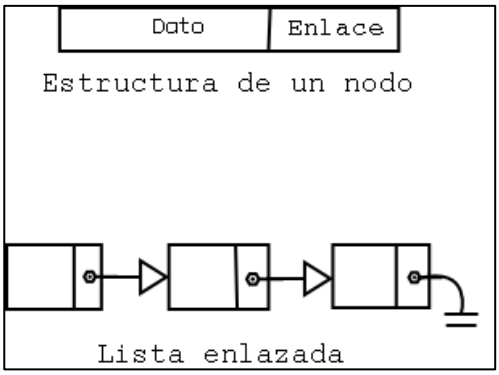

Figura 2. Diagrama de una lista enlazada simple.

Fuente: Diana Molina.

En la lista enlazada circular se uso para almacenar la información de cada matriz junto a sus atributos, esta lista también hace referencia a la lista simple enlazada donde se referencia con los datos que se compone la matriz.

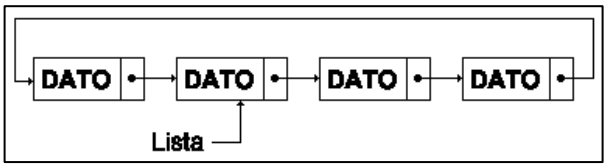

Figura 3. Diagrama de una lista circular enlazada.

Fuente: Alfonzo Rodríguez.

Con el paradigma de programación orientada a objetos se emulo los diferentes tipos de datos para almacenarlos en la estructura de datos correspondientes, esto abrió la posibilidad de trabajar más eficientemente y optimización de memoria del programa ya que se tienen varios datos y se necesitan agruparlos entre ellos para su manipulación.

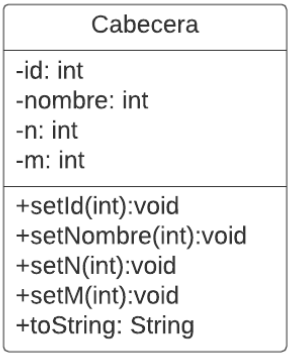

Figura 4. Modelo del tipo de dato Cabecera.

Fuente: Elaboración propia.

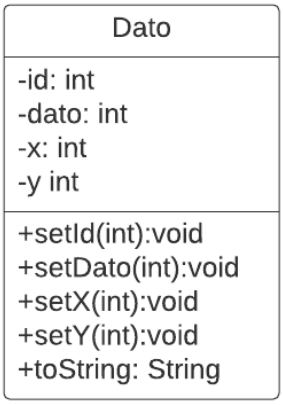

Figura 5. Modelo del tipo de dato Dato.

Fuente: Elaboración propia

Con las estructuras ya planteadas, se realizaron los siguientes algoritmos:

a. Extraer datos: este algoritmo obtiene toda la información que contiene el archivo XML proporcionado, identificando los nombres de las matrices junto a sus dimensiones y guardándola en una estructura de dato, también extrae la información de la matriz en cuestión y la almacena en otra estructura para

luego hacer uso de este, en caso de que las posiciones de los datos estén desordenados el programa se encarga de ordenarlos, para relacionar los nombres de las matrices con sus datos se hace a través del atributo id que ambas estructuras comparten.

b. Obtener patrones de acceso: este algoritmo convierte cada matriz en una matriz binaria, para poder el comportamiento de los datos.

c. Identificar grupos: teniendo las matrices de patrones de acceso se pueden identificar todas aquellas filas que se repiten, agrupándolas entre si para poder reducir la matriz, se obtiene información de las filas que se reducirán y guardando la información en una estructura.

d. Reducir matrices: con los grupos formados se hace referencia a la matriz que contiene los datos y se reduce sumando todas aquellas filas que se repiten según su patrón de acceso.

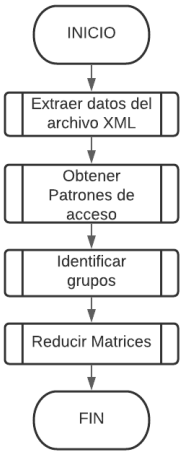

Figura 6. Diagrama general de los algoritmos.

Fuente: Elaboración propia

El último algoritmo provee la estructura del nuevo esquema para la base de datos, este se puede imprimir en un archivo con extensión y estructura XML, dando las nuevas matrices reducidas con sus grupos y frecuencias obtenidas.

El programa también da la opción de poder graficar las matrices obtenidas con el archivo de entrada, este hace uso de la herramienta graphviz para general la gráfica.

Conclusiones

Conocer e implementar bien las estructuras de datos permiten manejar de forma muy rápida y eficiente los datos almacenados.

La solución del problema da un ejemplo claro de cómo es que trabajan a nivel interno los motores de bases de datos para hacer consultas y agrupar datos según las condiciones dadas.

Se debe conocer a profundidad el lenguaje con el que se trabajara la solución ya que de estos dependen cuan accesible puede ser construir los algoritmos que determinaran la solución al problema, también es necesario conocer que paradigmas se pueden utilizar y en base a esto apegarse a una o varias para hacer lo menos tedioso posible el desarrollo de este.